Kotahi:
a new approach to JATS production

Dan Visel, Coko · *dbvisel@gmail.com*
Adam Hyde, Coko
Ben Whitmore, Coko

1. An introduction to Kotahi

Basic features

- Multiple workflows
- Multiple review models
- Document editing
- Realtime updates & chat
- Flexible approach to metadata

Basic features, continued

- Single-source methodology
- Versioning
- Exporting (to PDF, JATS, HTML, others)
- Integrations (ORCID, Crossref, Hypothes.is)
- Reports

Current users

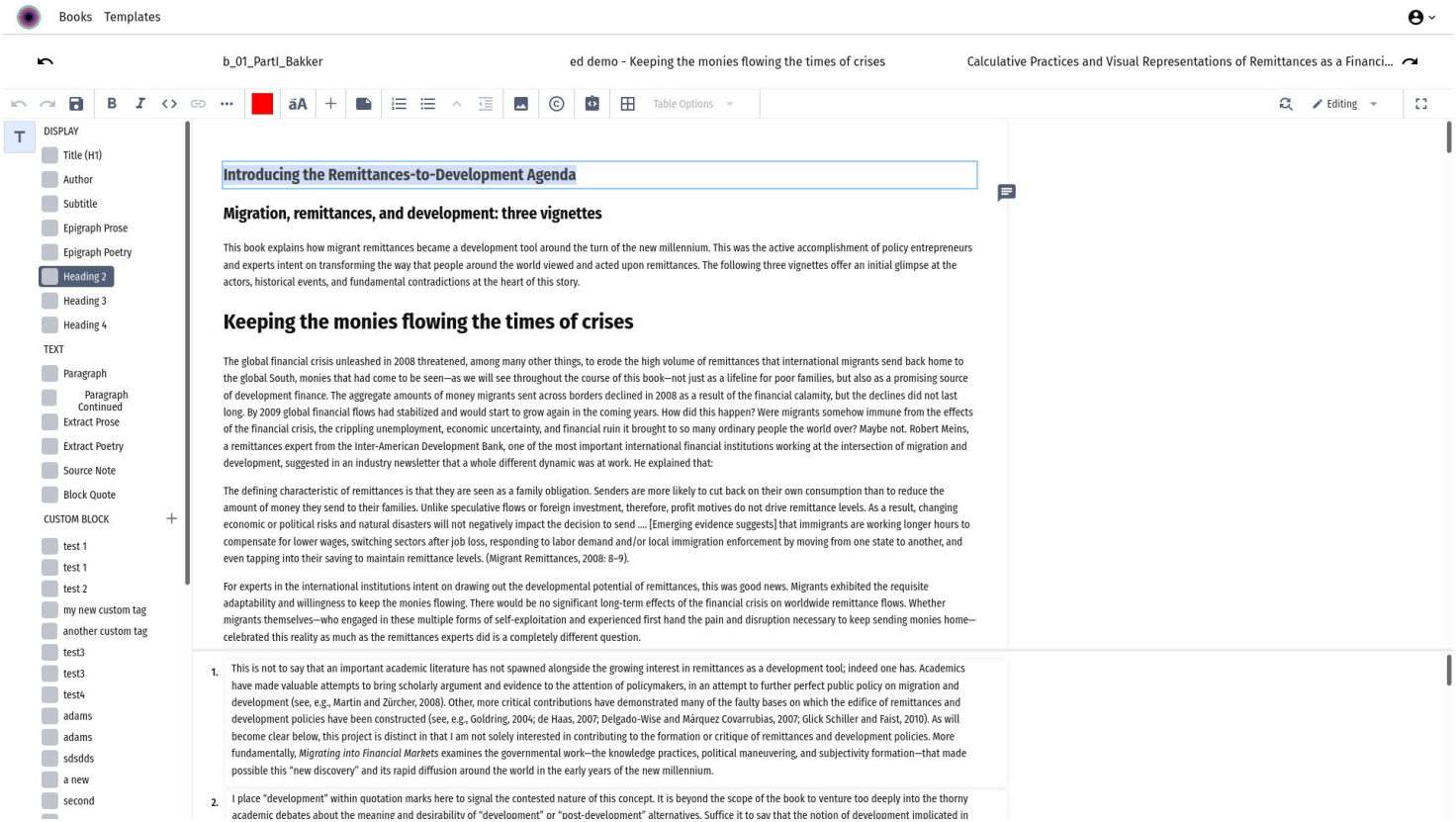
- eLife
- Amnet Systems
- JMIR
- Aperture Neuro

2. Background:
Coko & Kotahi’s infrastructure

The Coko Foundation

- The Coko Foundation, founded by Adam Hyde, makes publishing software
- Kotahi springs from Editoria, used to make books

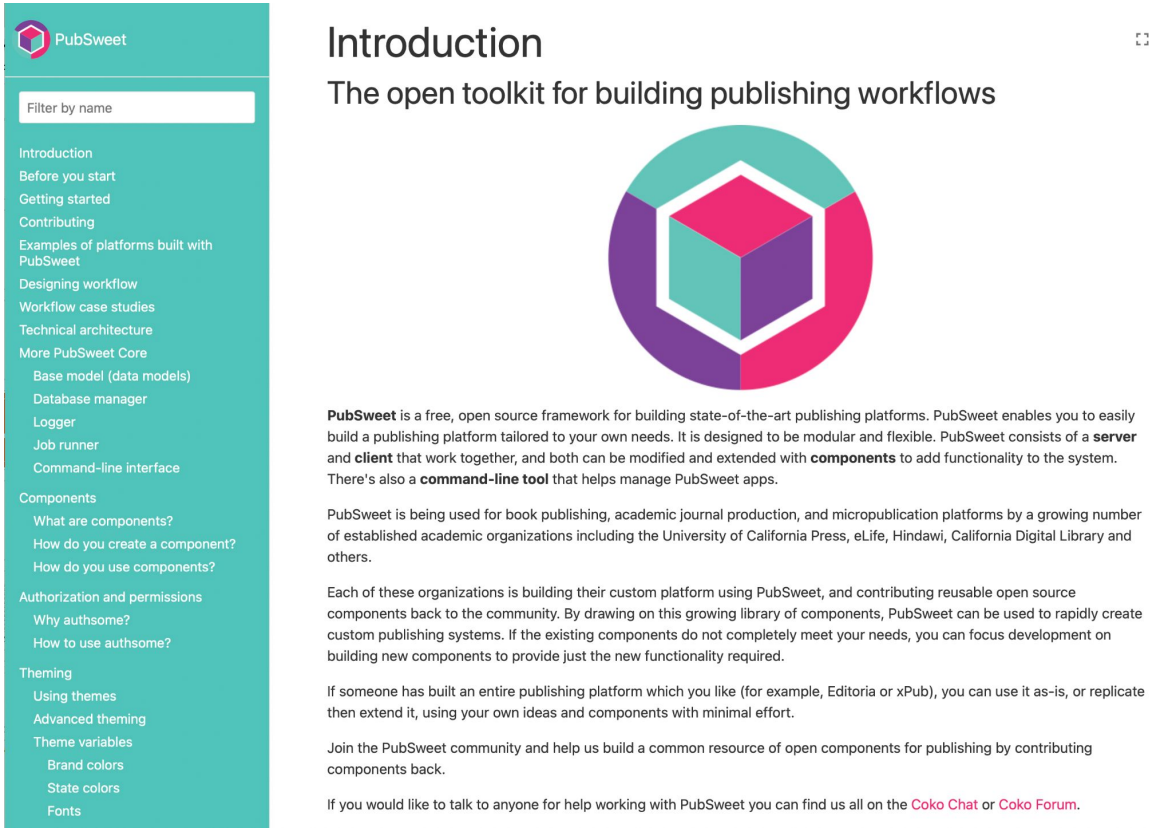
Editoria's book chooser



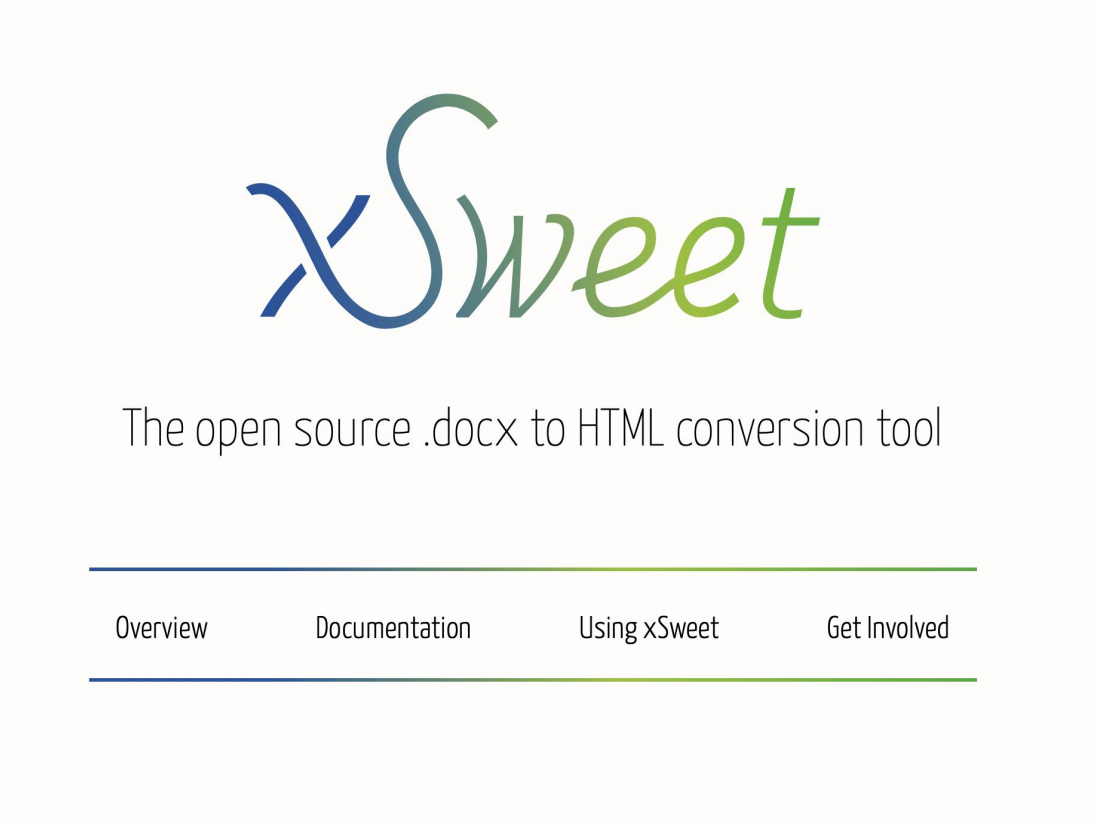
Editoria's editor – based on Wax, similar to Kotahi

Coko frameworks used by Kotahi

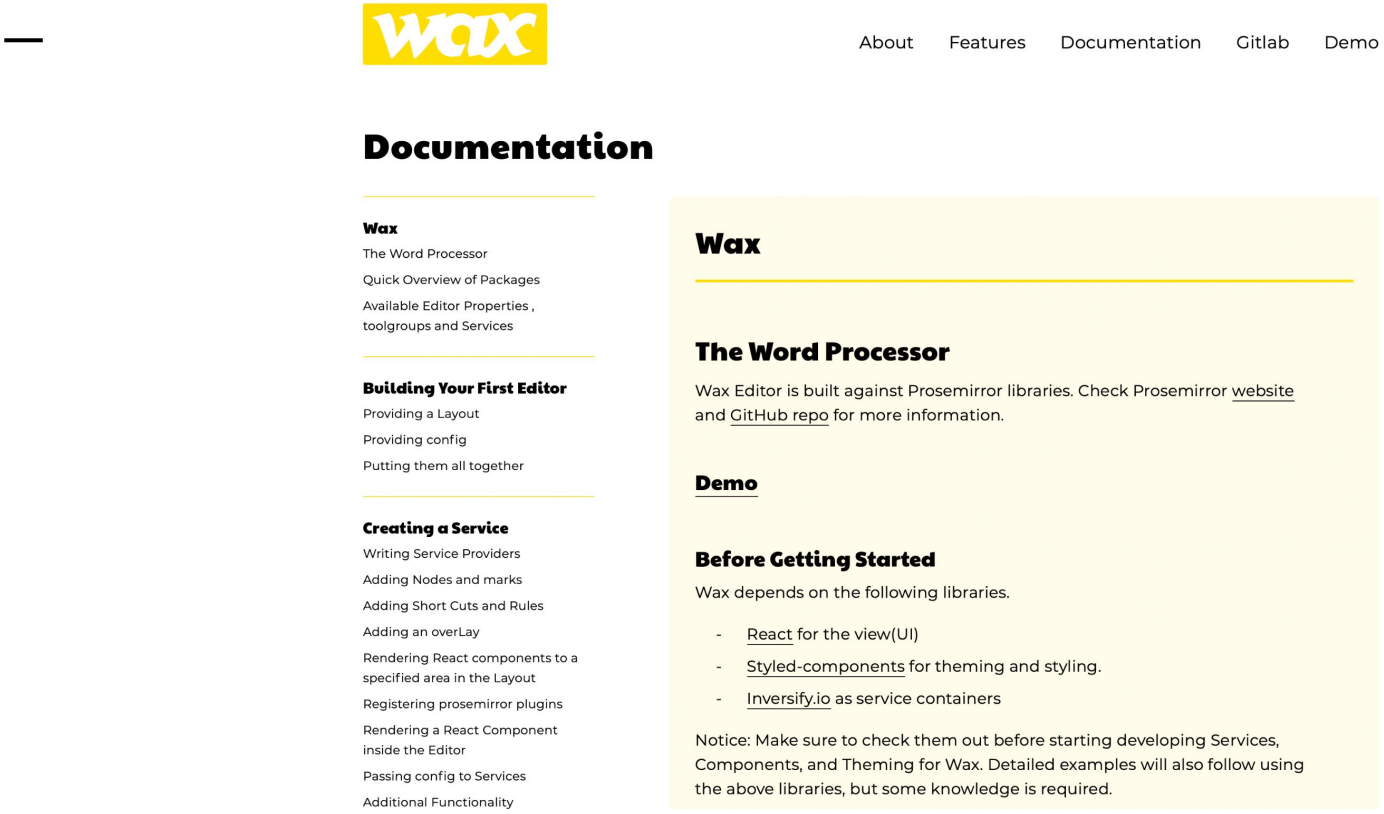
- **PubSweet**, component library
- **xSweet**, DOCX to HTML conversion
- **Wax**, web-based editor
- **Paged.js**, HTML to PDF conversion



PubSweet's homepage, documenting the component library



xSweet's homepage



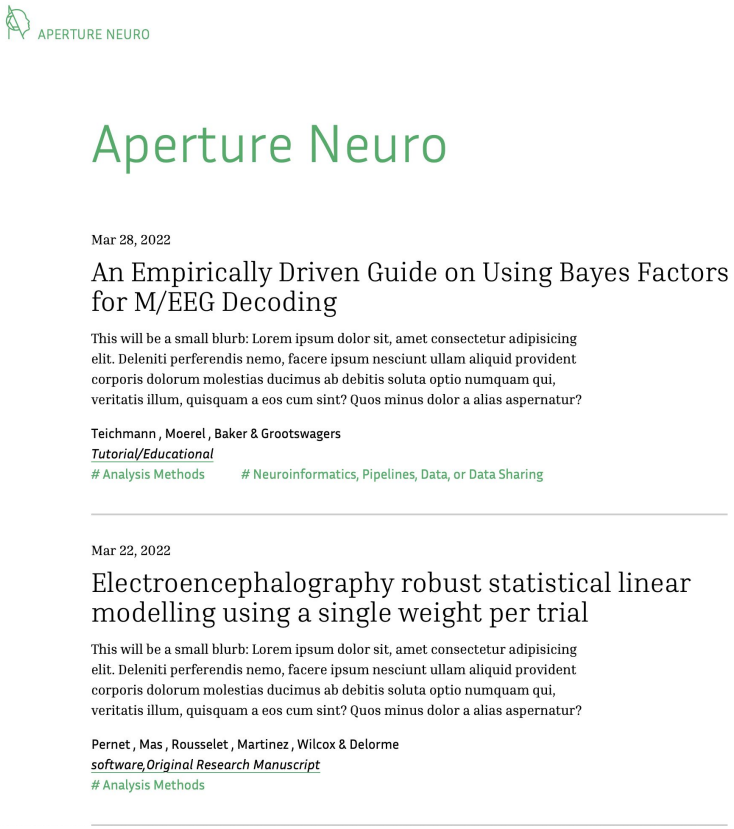
Wax's documentation

ABOUT DOCUMENTATION EXAMPLES JOURNAL GITLAB
paged.js
by Coko
OPEN SOURCE STANDARDS COMPLIANT COMMUNITY DRIVEN

Paged.js

Coko projects connected to Kotahi

- **Flax**, used as a web front end
- **Science Beam**, converts PDFs to JATS
- **Libero Editor**, another JATS editor, supported by Coko

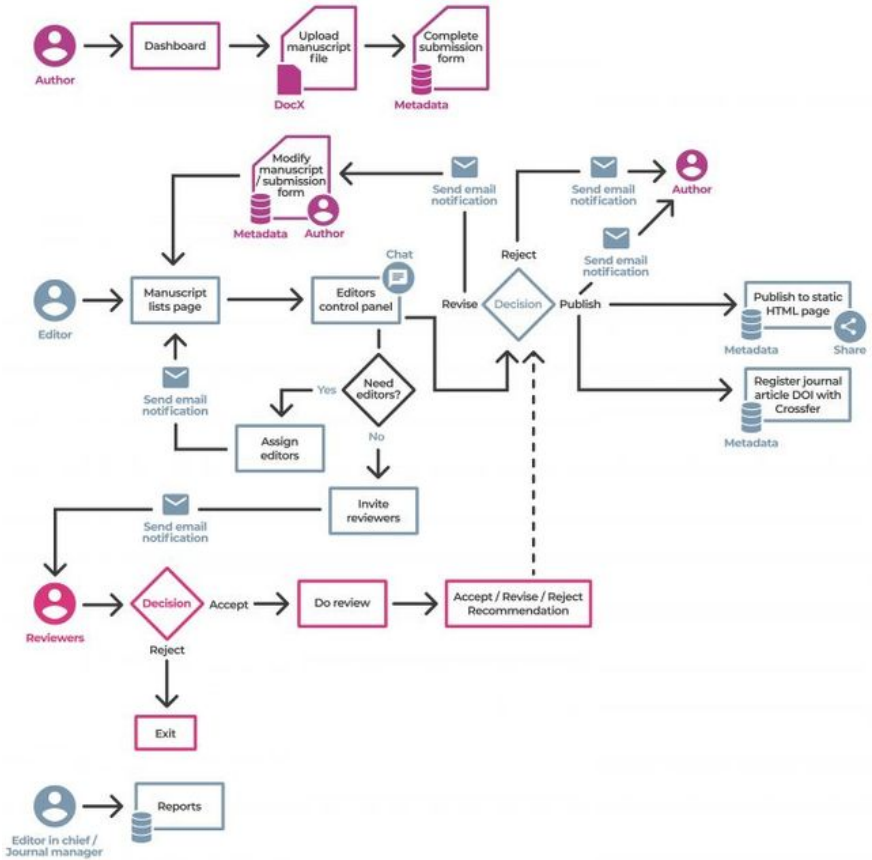


The Flax-generated page for Aperture Neuro, pulling content out of Kotahi

3. Key concepts

Workflow

- Publishers have wildly different workflows
- Kotahi is designed to accommodate individualized workflows
- Documents go through many states and travel from person to person
- Kotahi tries to keep as much of that in one app as possible



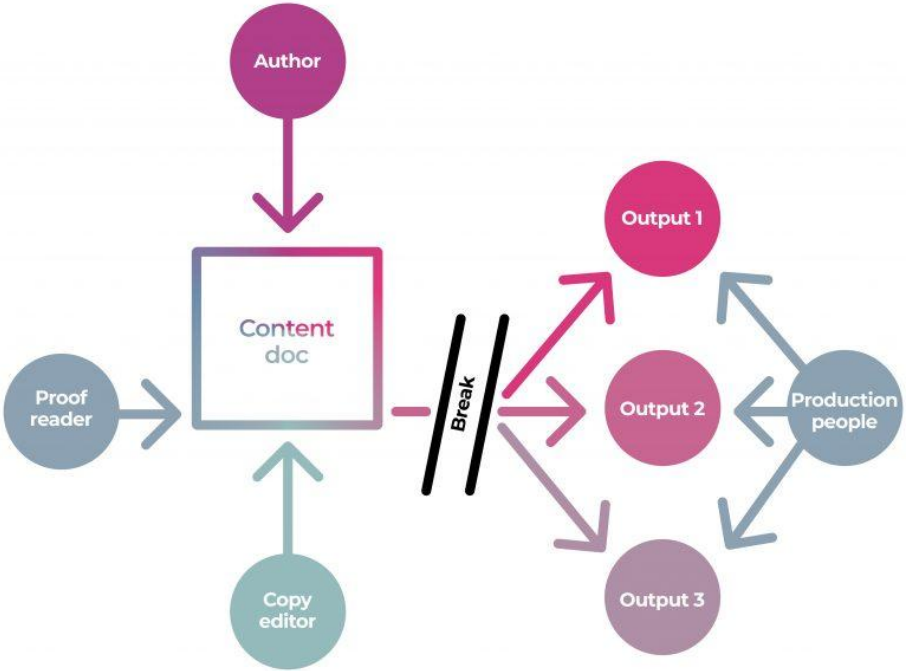
A workflow diagram for Kotahi in Aperture Neuro (design by Henrik van Leeuwen with Ryan Dix-Peek)

HTML as a source of truth

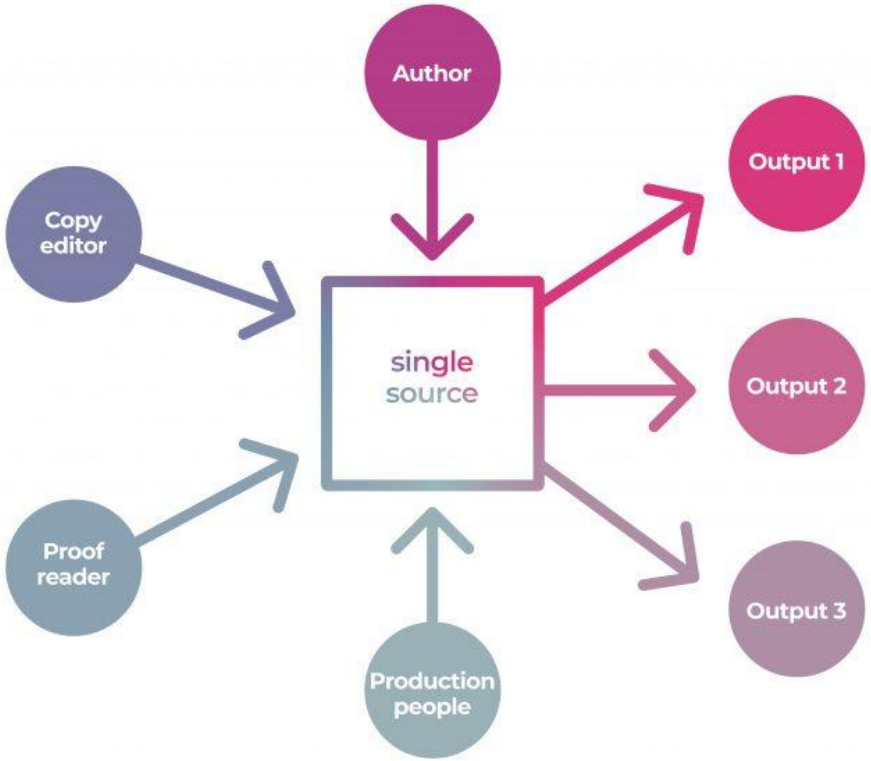
- Authors tend to use MS Word (or maybe Google Docs)
- We need to work around DOCX
- DOCX —> HTML can be done consistently (with xSweet)
- Editors, by and large, don’t think about structure.

Single-sourcing & the Kotahi model

- JATS is part of the workflow, not an endpoint
- A production editor is making JATS – but they're not necessarily a JATS expert
- Other people in the process could also be making JATS
- JATS might be updated



A typical publishing model

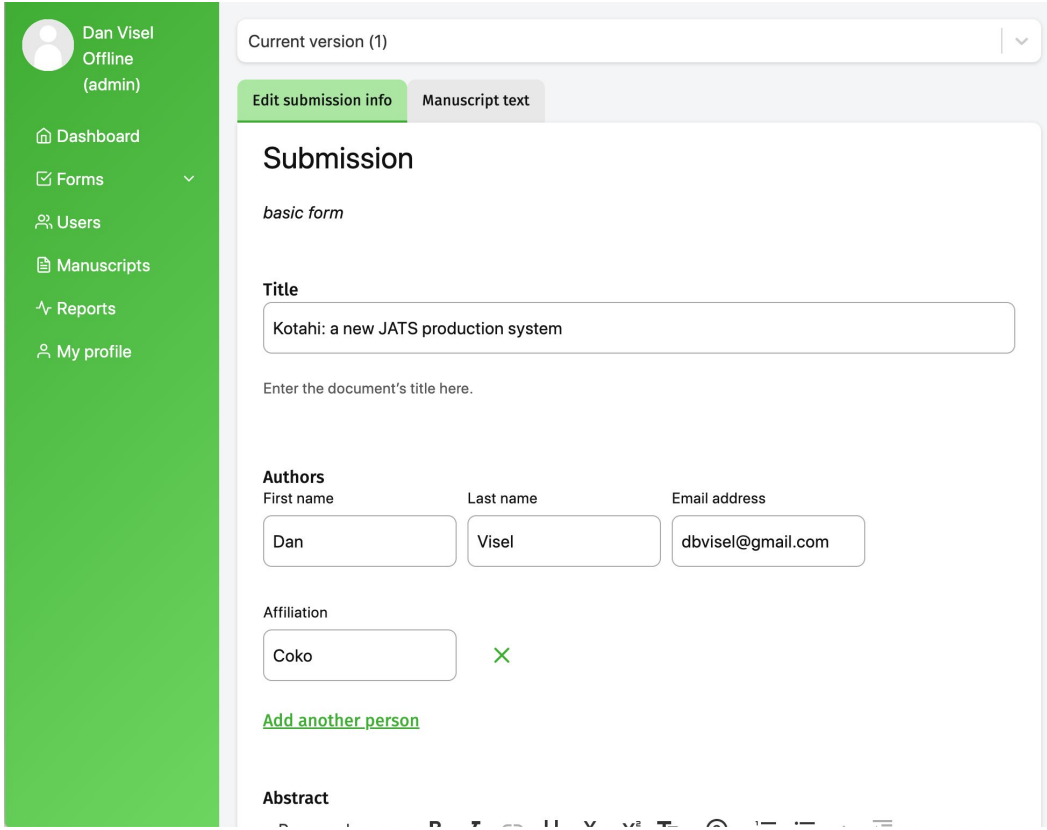


A single-source publishing model

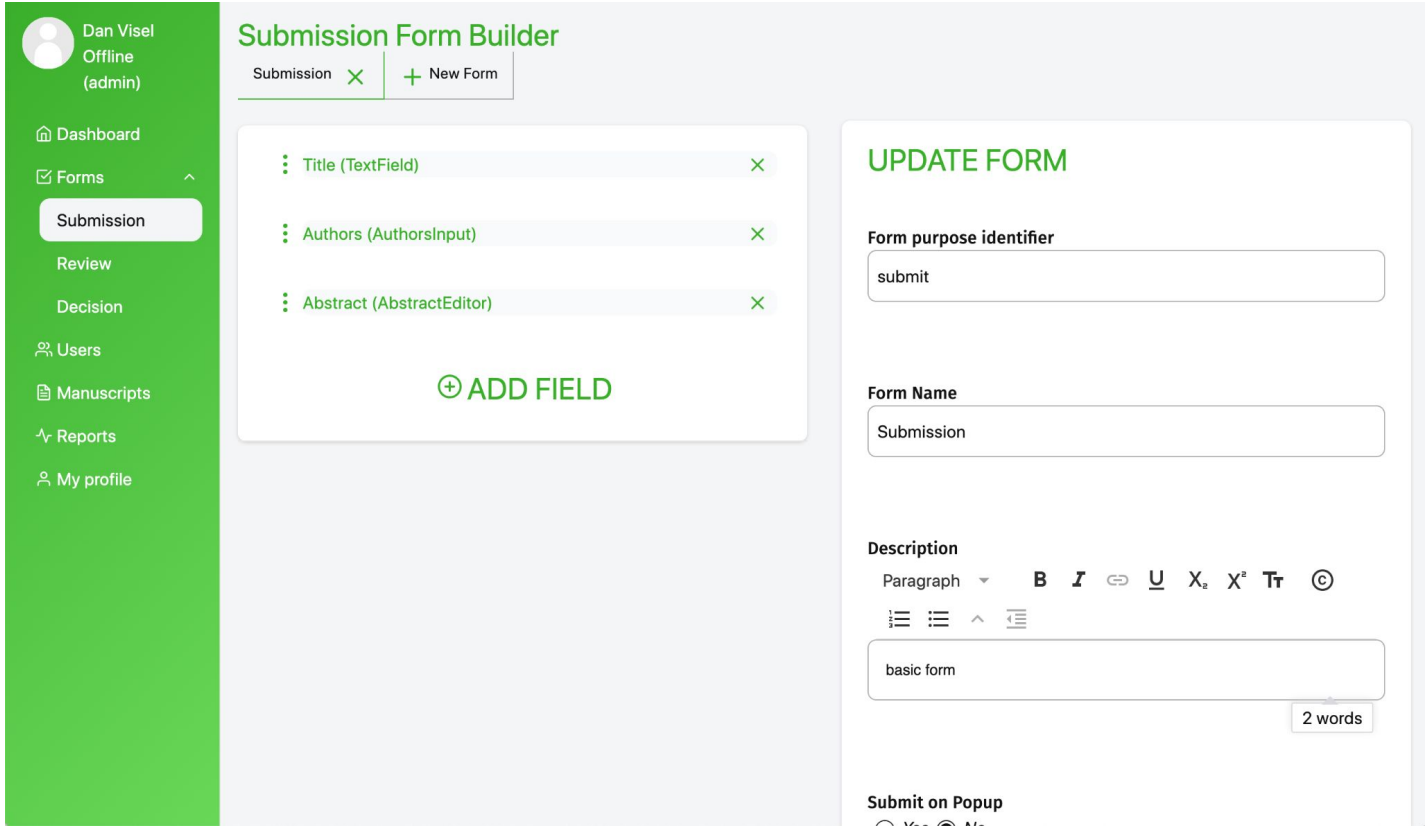
4. Documents in Kotahi

The form: control for metadata

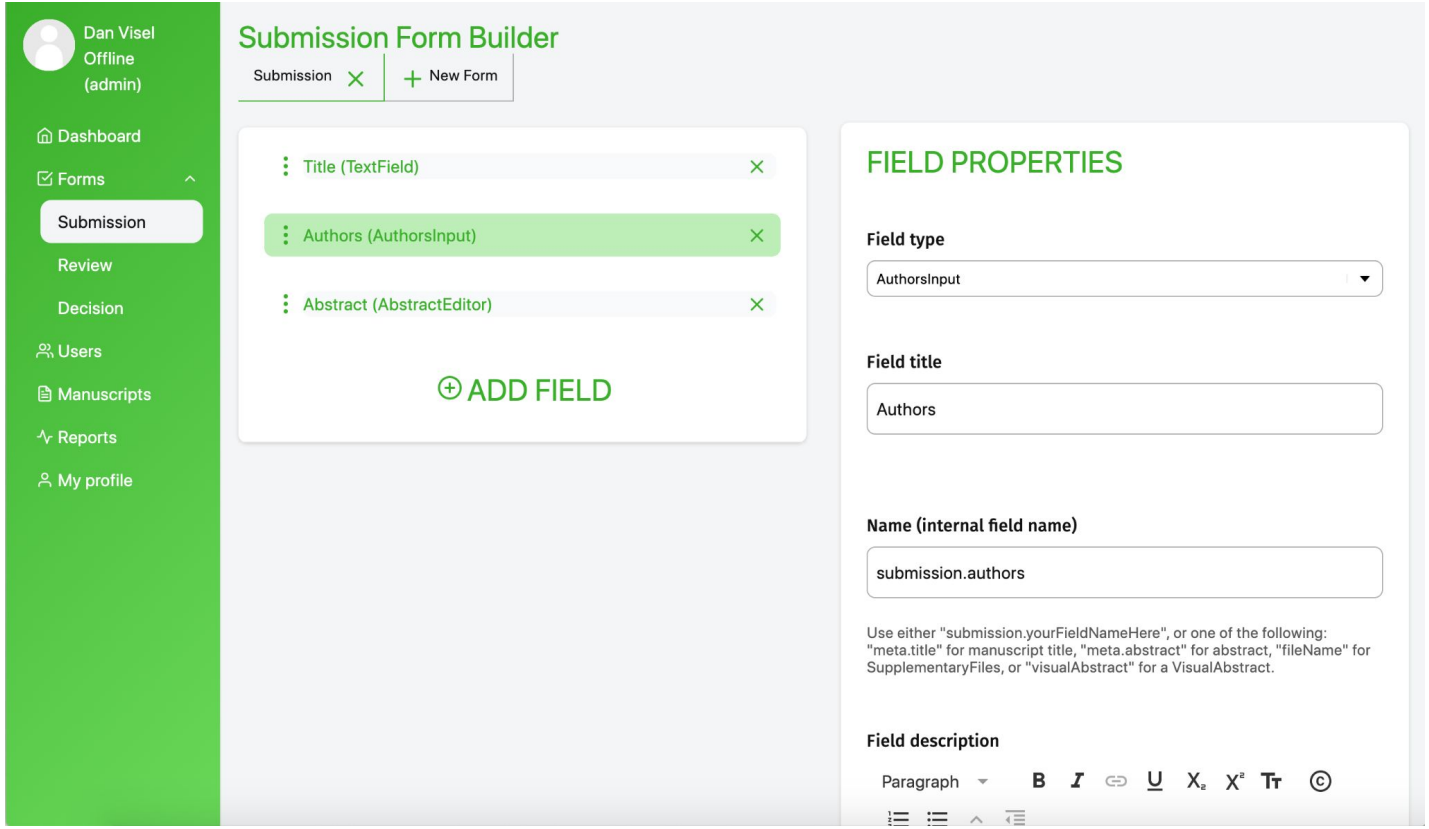
- Different points in the workflow need different kinds of metadata
- Some go into a document
- Some are for workflow (e.g. who is going to review this submission) and don’t end up as metadata
- Form could also include attachments (datasets, original files)



A sample form in Kotahi



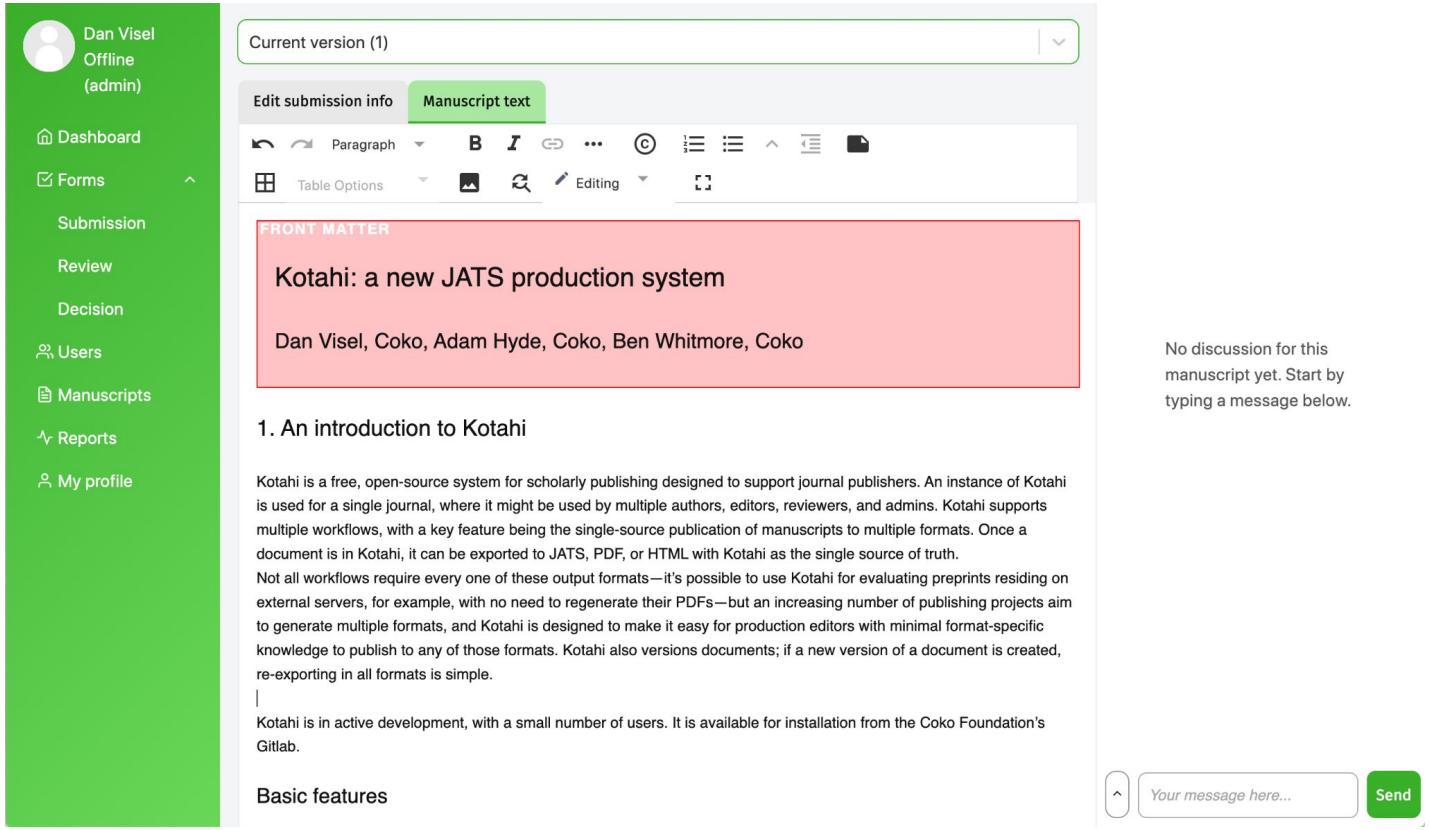
The Kotahi form builder: an instance can have multiple forms, each with their own fields



The field properties for a form: this is for a author's input (first & last names, email, affiliation)

Wax: where editing happens

- Wax is used in both the regular editor and the production editor
- Comments and changes are carried across
- JATS elements added in production are carried back into the regular editor, so changes can still be made.

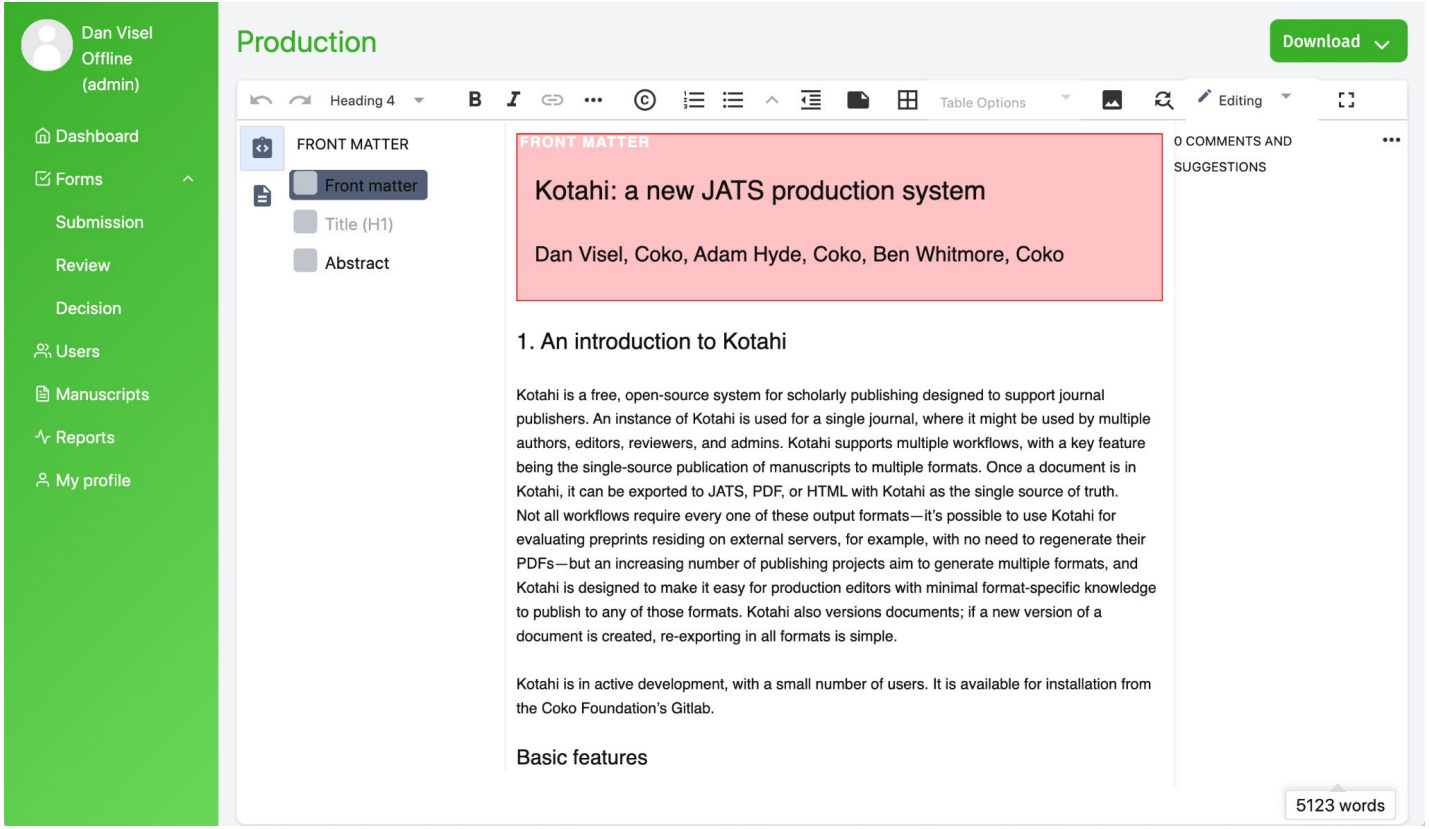


The general Wax editor in Kotahi, used by editors

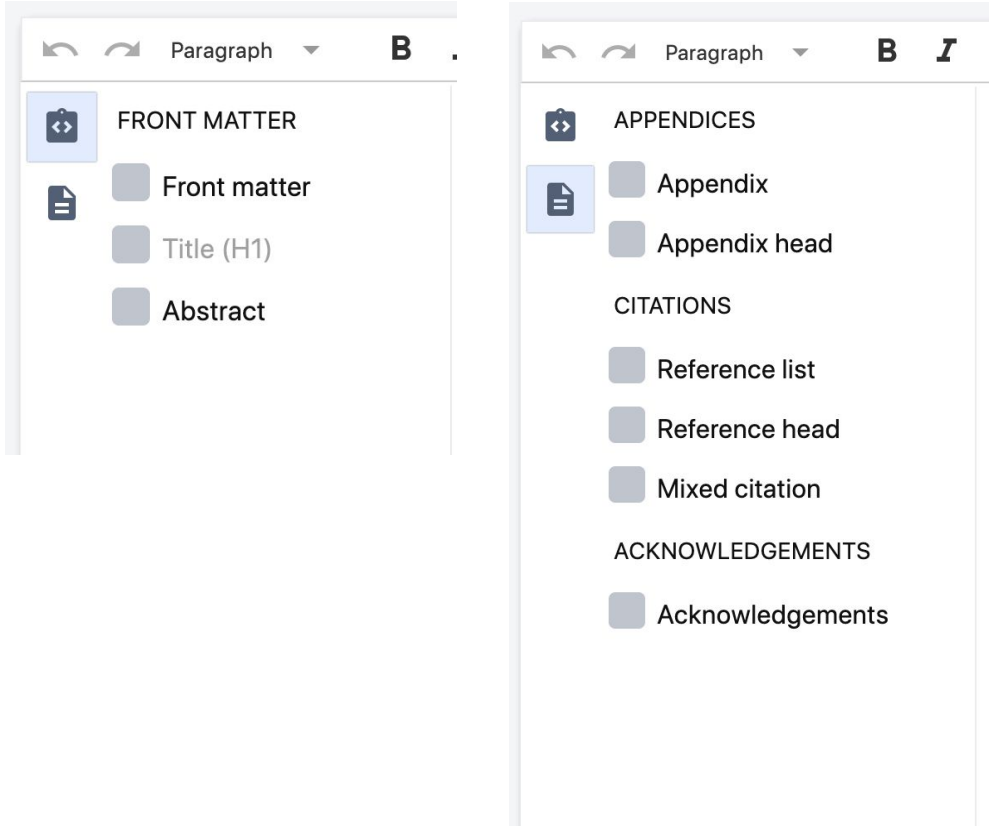
5. Making JATS in the production page

The production Wax editor

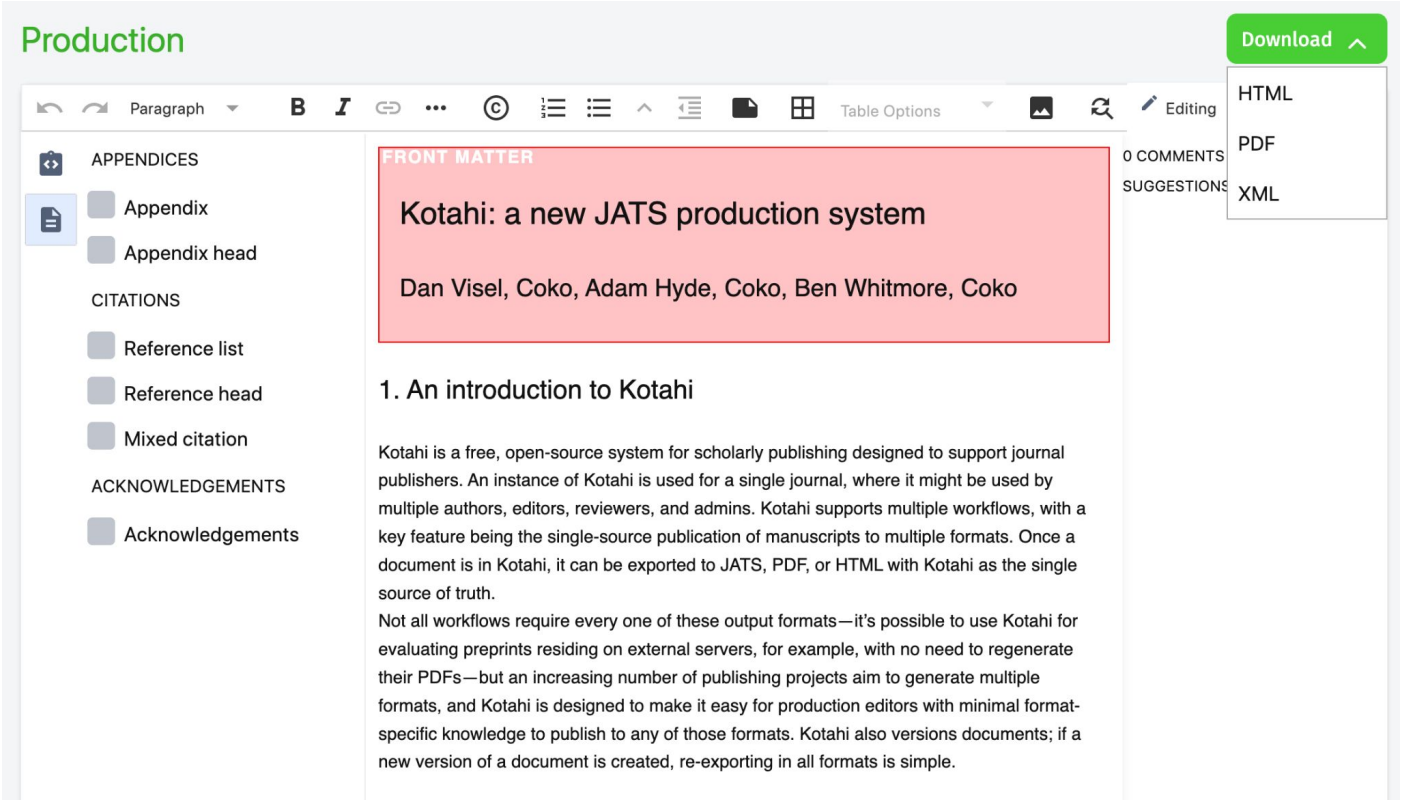
- Allows production editor to mark up pieces that need to go into JATS front or back matter
- What editor sees should be close to what's seen in the final PDF/HTML version
- Inserted JATS elements are just HTML tags, so they can go back and forth to the regular editor



The Wax editor on Kotahi production page, used by the production editor



Front matter and back matter elements in the production page editor’s drawer



In the upper right, a drop-down allows immediate download of HTML, PDF, and XML

Prepping for JATS

- Wax uses HTML, but it’s targeted at being converted to JATS using a closed taxonomy of tags and classes
- Internally, everything is relatively flat, but we can take structure implied through header and make a more nested XML-y structure

```
<h1>This is a top-level header</h1>
<p>This is content</p>
<h2>This is a second-level header.</h2>
<p>This is in the second level</p>
<h3>This is a third-level header</h3>
<h2>This is a second second-level header</h2>
```

How something might be structured in HTML – a flat structure

```
<sec>
  <title>This is a top-level header</title>
  <p>This is content</p>
  <sec>
    <title>This is a second-level header.</title>
    <p>This is in the second level</p>
    <sec>
      <title>This is a third-level header</title>
    </sec>
  </sec>
  <sec>
    <title>This is a second second-level header</title>
  </sec>
</sec>
```

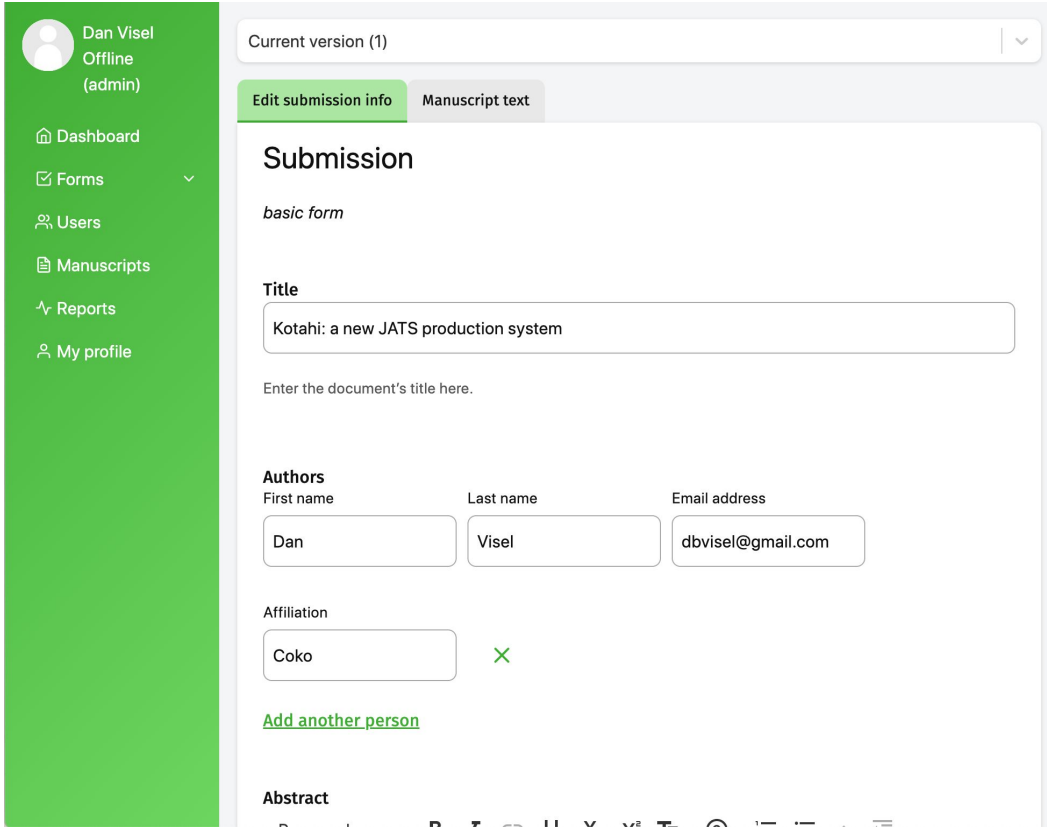
How that HTML is transformed into nested JATS tags

6. Exporting JATS

Where does the <front> come from?

<article-meta> comes from the form

<journal-meta> is set per instance



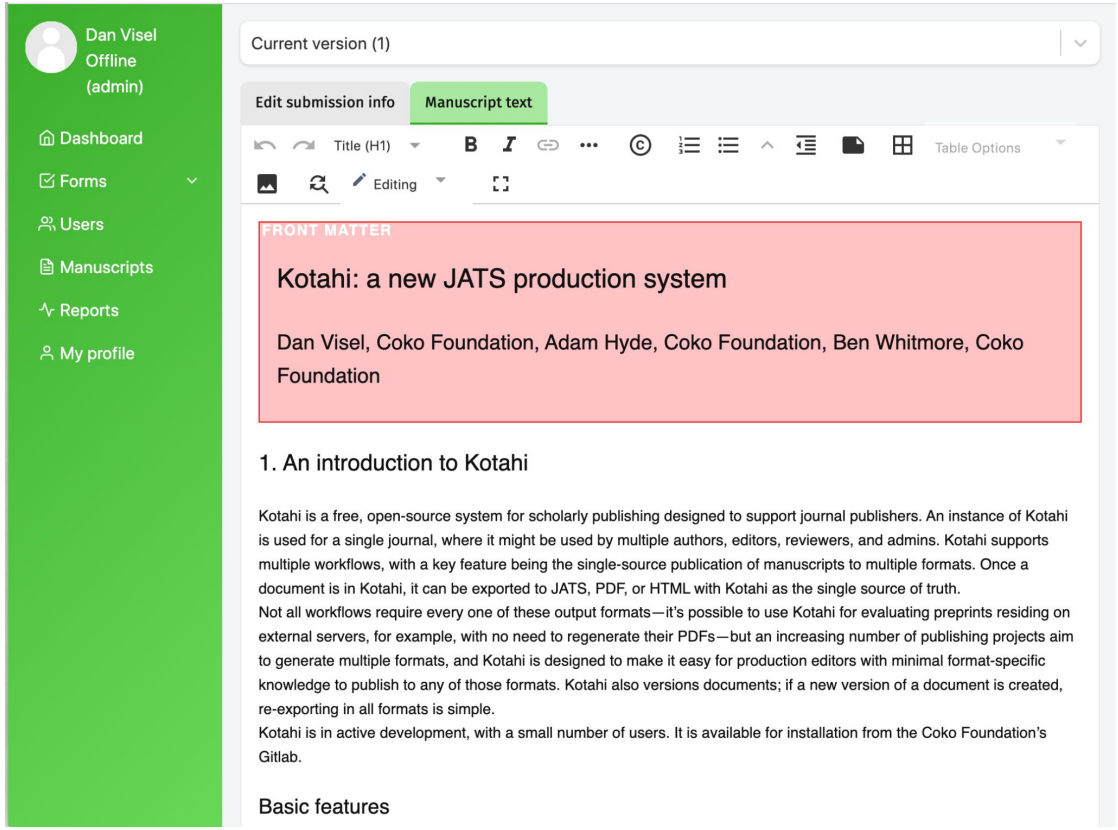
A sample form in Kotahi

```
▼<article xmlns:mml="http://www.w3.org/1998/Math/MathML" xmlns:xlink="http://www.w3.org/1999/xlink" xml:lang="en" dtd-version="1.3">
  ▼<front>
    ▼<journal-meta>
       <journal-id journal-id-type="pmc">BMJ</journal-id>
       <journal-id journal-id-type="publisher">BR MED J</journal-id>
      ▼<journal-title-group>
         <journal-title>Journal Title</journal-title>
         <abbrev-journal-title>Jour.Ti.</abbrev-journal-title>
       </journal-title-group>
       <issn publication-format="print">1063-777X</issn>
       <issn publication-format="electronic">1090-6517</issn>
       <publisher>elife</publisher>
     </journal-meta>
    ▼<article-meta>
      ▼<title-group>
         <article-title>Kotahi: a new JATS production system</article-title>
       </title-group>
      ▼<contrib-group>
        ▼<contrib contrib-type="author">
          ▼<name>
             <surname>Visel</surname>
             <given-names>Dan</given-names>
           </name>
           <xref ref-type="aff" rid="3d96c7ae-4e5f-4968-a58b-d3383fa1167e"/>
         </contrib>
       </contrib-group>
       <aff id="3d96c7ae-4e5f-4968-a58b-d3383fa1167e">Coko</aff>
      ▼<pub-date publication-format="print" date-type="pub" iso-8601-date="2022-4-22">
         <day>22</day>
         <month>4</month>
         <year>2022</year>
       </pub-date>
     </article-meta>
  </front>
```

Data from that form translated into a JATS front matter

Where does the <body> come from?

<body> comes from what the user sees in Wax – though some front & back matter material might be extracted.



The start of a document in Kotahi. Notice that some is wrapped in a front matter element

```
<section class="frontmatter">
  ::before
  <h1>Kotahi: a new JATS production system</h1>
  <h4>Dan Visel, Coko, Adam Hyde, Coko, Ben Whitmore, Coko</h4>
</section>
<h3>1. An introduction to Kotahi</h3>
<p class="paragraph">
  "Kotahi is a free, open-source system for scholarly publishing
  designed to support journal publishers. An instance of Kotahi is used
  for a single journal, where it might be used by multiple authors,
  editors, reviewers, and admins. Kotahi supports multiple workflows,
  with a key feature being the single-source publication of manuscripts
  to multiple formats. Once a document is in Kotahi, it can be exported
  to JATS, PDF, or HTML with Kotahi as the single source of truth."
</p>
<p class="paragraph">
  "Not all workflows require every one of these output formats—it's
  possible to use Kotahi for evaluating preprints residing on external
  servers, for example, with no need to regenerate their PDFs—but an
  increasing number of publishing projects aim to generate multiple
  formats, and Kotahi is designed to make it easy for production editors
  with minimal format-specific knowledge to publish to any of those
  formats. Kotahi also versions documents; if a new version of a
  document is created, re-exporting in all formats is simple."
</p>
<p class="paragraph">…</p>
<p class="paragraph">…</p>
```

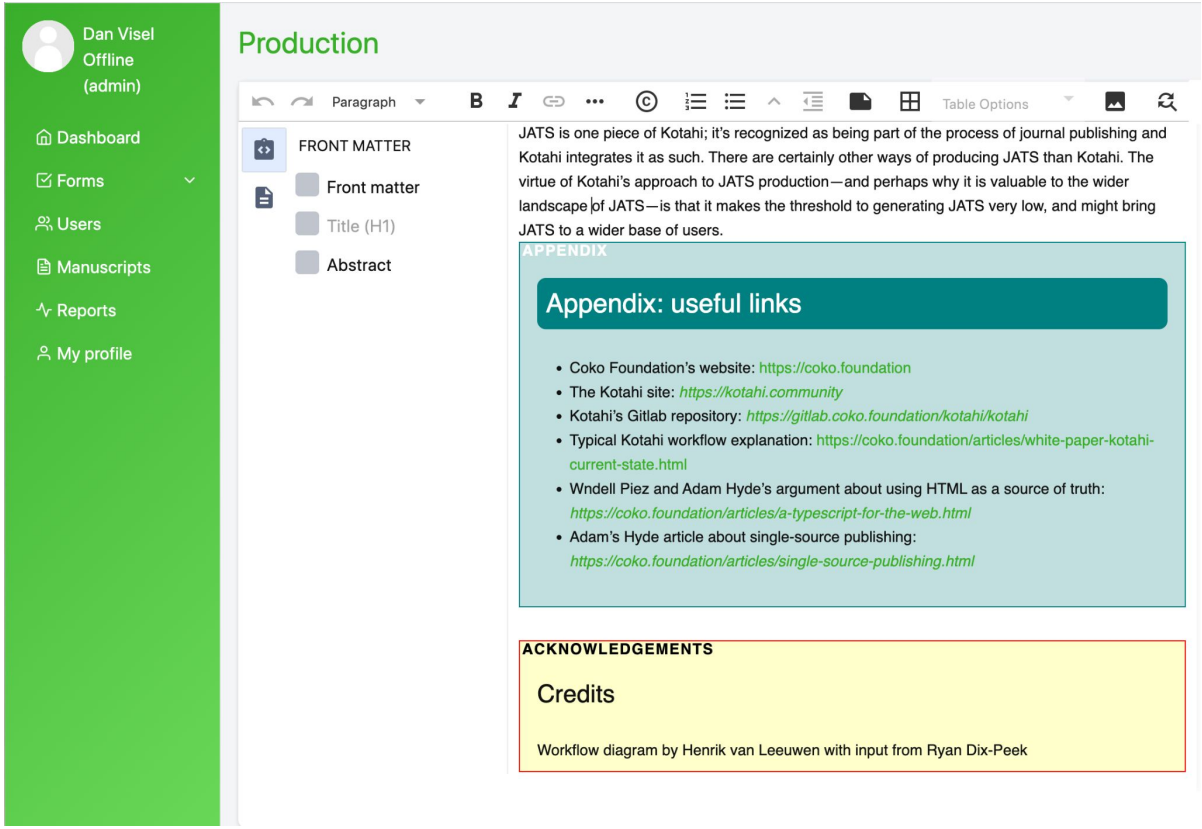
How that's represented internally as HTML

```
▼<body>
  ▼<sec>
    <title>1. An introduction to Kotahi</title>
    <p>Kotahi is a free, open-source system for scholarly publishing designed to support journal publishers. An instance
    of Kotahi is used for a single journal, where it might be used by multiple authors, editors, reviewers, and admins.
    Kotahi supports multiple workflows, with a key feature being the single-source publication of manuscripts to multiple
    formats. Once a document is in Kotahi, it can be exported to JATS, PDF, or HTML with Kotahi as the single source of
    truth.</p>
    <p>Not all workflows require every one of these output formats—it's possible to use Kotahi for evaluating preprints
    residing on external servers, for example, with no need to regenerate their PDFs—but an increasing number of
    publishing projects aim to generate multiple formats, and Kotahi is designed to make it easy for production editors
    with minimal format-specific knowledge to publish to any of those formats. Kotahi also versions documents; if a new
    version of a document is created, re-exporting in all formats is simple.</p>
    <p>Kotahi is in active development, with a small number of users. It is available for installation from the Coko
    Foundation's Gitlab.</p>
  ▼<sec>
      <title>Basic features</title>
      <p>Main features include:</p>
    ▼<list list-type="bullet">
      ▼<list-item>
        ▼<p>
            <italic>Workflows:</italic>
            Kotahi supports multiple workflows including PRC (Publish, Review, Curate) and the submission, peer review,
            and publishing of everything from preprints, micropubs, journals, and books to exciting new types of
            publishing.
          </p>
        </list-item>
      ▼<list-item>
        ▼<p>
            <italic>Review models:</italic>
            Kotahi enables publishers to use a number of different review models: open, blind, or double blind. Reviewers
            can have individual reviews or collaborate on a shared review.
          </p>
        </list-item>
      ▼<list-item>
        ▼<p>
            <italic>Editing:</italic>
            Kotahi is designed around the idea of editing and authoring articles in the browser, using Wax, Coko's full-
            featured web-based word processor.
          </p>
        </list-item>
```

The start of that document turned into JATS. The front matter material is gone.

Where does the <back> come from?

<back> comes from specially tagged elements in Wax – and possibly some elements from the form.



The end of a document in Kotahi. There's an appendix and acknowledgments.

What that looks like as HTML internally. Pseudo-elements provide editorial hints via CSS

```
<back>
  <ack>
    <sec>
      <title>Credits</title>
      <p>Workflow diagram by Henrik van Leeuwen with input from Ryan Dix-Peek</p>
    </sec>
  </ack>
  <app-group>
    <app id="app-0">
      <title>Appendix: useful links</title>
      <list list-type="bullet">
        <list-item>
          <p>
            Coko Foundation’s website:
            <ext-link xmlns:xlink="http://www.w3.org/1999/xlink" ext-link-type="uri"
            xlink:href="https://coko.foundation">https://coko.foundation</ext-link>
          </p>
        </list-item>
        <list-item>
          <p>
            The Kotahi site:
            <italic>
              <ext-link xmlns:xlink="http://www.w3.org/1999/xlink" ext-link-type="uri"
              xlink:href="https://kotahi.community">https://kotahi.community</ext-link>
            </italic>
          </p>
        </list-item>
        <list-item>
          <p>
            Kotahi’s Gitlab repository:
            <italic>
              <ext-link xmlns:xlink="http://www.w3.org/1999/xlink" ext-link-type="uri"
              xlink:href="https://gitlab.coko.foundation/kotahi/kotahi">https://gitlab.coko.foundation/kotahi/kotahi</ext-link>
            </italic>
          </p>
        </list-item>
        <list-item>
          <p>
            Typical Kotahi workflow explanation:
            <ext-link xmlns:xlink="http://www.w3.org/1999/xlink" ext-link-type="uri"
            xlink:href="https://coko.foundation/articles/white-paper-kotahi-current-
            state.html">https://coko.foundation/articles/white-paper-kotahi-current-state.html</ext-link>
          </p>
        </list-item>
        <list-item>
          <p>
            Wndell Piez and Adam Hyde’s argument about using HTML as a source of truth:
            <italic>
              <ext-link xmlns:xlink="http://www.w3.org/1999/xlink" ext-link-type="uri"
              xlink:href="https://coko.foundation/articles/a-typescript-for-the-web.html">https://coko.foundation/articles/a-
```

And those elements go into the <back> tag in JATS

Validation

- Exported JATS is validated both as XML and against the JATS schema
- This is primarily valuable to developers: if Kotahi is properly set up, it should not be possible to export invalid JATS.
- This is because of the constrained approach to functionality that Kotahi takes.

7. Expansion into the future

Towards a more WYSIWYG approach

- Metadata is currently in the form rather than the Wax editor.
- Moving that content into Wax – in the form of structured content – would allow editors to directly edit metadata as it would appear in the printed version.

Citations

- Kotahi’s citations are currently very simple
- A citation maker might be attached to allow manipulation of citations

Math

- Math in Kotahi is currently MathJax
- Math is exported as SVGs and MathML
- Different clients seem to want different approaches (so editors don’t need to know LaTex); a WYSIWYG editor is a possibility
- MathJax supports mhchem for typesetting chemistry; support could be included

Different ways of exporting

- Export is currently on demand
- More persistent PDF/JATS might be imagined
- Publication should automatically generate PDF/JATS for different endpoints
- Figuring out how people want to consume the final product

Making JATS friendly

- Kotahi doesn’t let you create invalid JATS
- Visually explaining to users why you can’t have two <abstract> sections (for example) – on export to JATS, only one of them will be there.

8. Conclusion

JATS for everyone

- Kotahi wants to let everyone create JATS without necessarily knowing JATS
- A rising tide lifts all boats?

Thanks!